

# **USB Feature Specification: Interface Power Management**

**MICROSOFT CORPORATION  
INTEL CORPORATION**

**Revision 1.0**

## Scope of this Revision

Converted from a review request (rr) format to new format for USB feature specification; no content changes.

## Revision History

Revision	Issue Date	Comments
1.0rc4d	7/27/99	Intel and Microsoft did final in-house review and required edits for clarification; updated page headers and footers and Table of Contents
1.0rc4c	5/13/99	Edited Interface Power Management spec; made the changes agreed to at the last Common Class Working Group meeting.
1.0rc4b	3/23/99	Split specification into two specifications: one for interface power management and the other for device power budgeting. This enables the interface power management spec to move along on a faster track than the device power budgeting spec, where there are still many significant issues.
1.0rc4a	2/2/99	Made general changes suggested by Terry Moore (MCCI) and rewrote section 4.3.4, using Terry's ideas.
1.0rc4	1/11/99	Reorganized to make more usable; deleted redundancy; fixed bugs; reformatted prose into tables.
1.0rc3	12/7/98	Incorporated all feedback from Common Class CWG meeting in Tigard, OR
1.0rc2	12/1/98	In section 5.1ff, clarified rule that if a configuration power descriptor is present, there must be a power descriptor for each interface (including alternate interfaces). Rewrote section 4.3. Eliminated obsolete description of "USB System Software Model." Added subsection to section 4.3 that describes how wake-up enabled interface on bus-powered device must resume from USB Suspend. Added minor changes to sections 2 and 3 to make clear scope of this feature is both device and interface power management.
1.0rc1	8/12/98	Expanded Contributors List to include all the people that contributed to the specification. Specified values for new feature selectors in section 6.1. Clarified the rules for including power descriptors on a device in section 5.
1.0rc	7/14/98	Accepted all changes in v0.9rd specification to create v1.0rc
0.9rd	6/10/98	Changes suggested by the Common Class Working Group at face-to-face meeting on Long Island, New York, are shown with revision marks in this v0.9rd release of the feature specification
0.9a	1/23/98	Added power consumption and user latency fields to Power Descriptor. Eliminated Appendix
0.8b	1/28/98	Made numerous changes suggested by Common Class Working Group at face-to-face meeting in Atlanta, 1/26/98.
0.7	12/2/97	Converted from a review request (rr) format to new format for USB feature specification; no content changes

## Contributors

John Dunn	Microsoft Corporation
Mark Williams	Microsoft Corporation
Ken Ray	Microsoft Corporation
Dave Lawrence	GTDI
John Howard	Intel Corporation
Geert Knapen	Philips
Tony Tran	CMD
Dominique Lallement	MGE
Kenny Richards	3Com
Ajay Shah	Phoenix Technologies
Steve Chang	KC Technology
Steve McGowan	Intel Corporation
Bruce Ramsland	AMP
Janet Schank	Digital
Mats Webjorn	Universal Access
Chuck Brebanek	Intel Corporation
Brad Hosler	Intel Corporation
Terry Moore	MCCI

Universal Serial Bus Class Definitions  
Copyright © 1997- 2001 by USB Implementers Forum  
All rights reserved.

### INTELLECTUAL PROPERTY DISCLAIMER

**THIS SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE.**

**A LICENSE IS HEREBY GRANTED TO REPRODUCE AND DISTRIBUTE THIS SPECIFICATION FOR INTERNAL USE ONLY. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY OTHER INTELLECTUAL PROPERTY RIGHTS IS GRANTED OR INTENDED HEREBY.**

**AUTHORS OF THIS SPECIFICATION DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF PROPRIETARY RIGHTS, RELATING TO IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. AUTHORS OF THIS SPECIFICATION ALSO DO NOT WARRANT OR REPRESENT THAT SUCH IMPLEMENTATION(S) WILL NOT INFRINGE SUCH RIGHTS.**

All product names are trademarks, registered trademarks, or servicemarks of their respective owners.

*Please send comments via electronic mail to [ccscomments@usb.org](mailto:ccscomments@usb.org)*

# Table of Contents

<b>1.</b>	<b>INTRODUCTION .....</b>	<b>1</b>
1.1	Purpose.....	1
1.2	Scope.....	1
1.3	1.3 Related Documents.....	1
1.4	Terms and Abbreviations .....	2
<b>2.</b>	<b>MANAGEMENT OVERVIEW .....</b>	<b>3</b>
<b>3.</b>	<b>3. FUNCTIONAL CHARACTERISTICS .....</b>	<b>4</b>
3.1	Feature Requirements .....	4
3.2	Interface Power States .....	4
3.3	Backward Compatibility with Existing Devices .....	5
3.3.1	Host Handling of Devices without Power Descriptors.....	5
3.3.2	Device Handling of Request Extensions .....	5
<b>4.</b>	<b>OPERATIONAL MODEL.....</b>	<b>7</b>
4.1	How Device Interfaces Respond to Host Power Management Requests .....	7
4.2	General Device Behavior in the Different Power States .....	7
4.3	How Device Interfaces Respond to USB Bus Power Events .....	8
<b>5.</b>	<b>DESCRIPTORS.....</b>	<b>10</b>
5.1	Interface Power Descriptor .....	10
5.1.1	Interface Power Descriptor Placement .....	10
5.1.2	Interface Power Descriptor Format .....	10
<b>6.</b>	<b>REQUESTS.....</b>	<b>11</b>
6.1	Set Feature and Clear Feature Selectors.....	11
6.2	Get Status Request Extensions.....	11

# 1. Introduction

## 1.1 Purpose

This document describes requirements and specifications for providing power management to an interface on a USB device. Achieving maximum system-wide power savings on PCs is an important design goal in light of the Energy Star™ program, the Five Watt PC goal, and the goal of maximum battery life on mobile computers. With these ambitious low power consumption goals, both PC peripheral and PC platform designers must take power management into account in all aspects of their designs.

USB devices must contribute all they can to system-wide power savings. The *Universal Serial Bus Specification*, which requires all USB devices to implement a low-power Suspend mode (controlled over the USB bus), goes a long way in contributing to system-wide power savings. With the addition of a USB interface power management feature, USB devices can contribute even more power savings.

## 1.2 Scope

This Common Class feature specification uses the OnNow Power Management initiative architecture and device power management model for PC peripherals. For more information about the OnNow power management model, see section 3.

The information in this document

- Enables USB device designers and developers to participate in a device power management scheme that reduces system-wide power consumption.

USB devices are not required to implement any of the capabilities described in this document.

## 1.3 1.3 Related Documents

*USB Specification, Version 1.1*, available at <http://www.usb.org>.

*USB Common Class Specification, Version 1.0*, available at <http://www.usb.org/developers>.

*Advanced Configuration and Power (ACPI) Specification, Version 1.0*, available  
<http://www.microsoft.com/hwdev>.

## 1.4 Terms and Abbreviations

Energy Star <sup>TM</sup> program	A United States Federal Government program that encourages PC platform makers and monitor makers to produce PCs and monitors that have low power consumption modes. Studies have shown that organizations with a large number of PCs can save significant amounts of money by installing PCs and monitors that have power management features.
Five Watt PC	Design goal set by large PC makers, along with PC component makers. The PC appears off but is quickly ready to use like a VCR or TV. The PC can respond to external events such as a phone ring, arrival of a packet over the LAN, or receipt of signal from a remote control.

## 2. Management Overview

This section is an overview of the contents of this document and provides a brief summary of each of the subsequent sections. It does not establish any requirements or guidelines.

In general, the *Universal Serial Bus Specification* contains the device requirements that enable the USB host software to contribute to system-wide power management (power savings and wake-up from system sleeping states). These device requirements are:

- A device can draw no more than 500 mA from the USB bus.
- A device can draw no more than 100 mA from the USB bus before being configured by the host software.
- A device can draw no more than 500 PA from the USB bus when it is suspended. The only exception is a high-power device configured for high-power and enabled as a remote wakeup source, which may draw up to 2.5mA while the device is suspended (for more information, see section 7.2.3 of the *USB Specification, Version 1.1*).
- A suspended device can signal a host that may also be suspended. This notifies the host that it should resume from its suspended mode, if necessary, and service the external event that triggered the suspended USB device to signal the host.

(Note: These bullet items summarize the USB power distribution model given in section 7.2 of the *Universal Serial Bus Specification* and the remote wakeup overview given in section 9.2.5.2 of the *Universal Serial Bus Specification*).

This common class feature specification extends power management to the interface level on a USB device. This allows

- Devices and functions that follow this specification can take advantage of new current limits for suspended and resuming devices (up to 100mA). For more information, see section 4.3.
- Each interface on a USB device to be treated as a separate function and managed/controlled by a separate device driver, and operate seamlessly in a power managed system.

Section 3, Functional Characteristics, lists the requirements this interface power management specification must meet, defines all the power states a USB interface can implement, and specifies how backward compatibility with USB interfaces that do not implement this feature is maintained.

Section 4, Operational Model, describes how a device that implements this optional feature interacts with the host and to events on the USB bus.

Section 5, Descriptors, defines a new interface power descriptor and defines the rules of placement for interface power descriptors in a Configuration descriptor for the device.

Section 6, Requests, defines new feature selectors for the Set Feature and Clear Feature requests that an interface must respond to if it implements this optional feature. Section 6 also defines new information returned in response to a Get Status request directed to an interface.

### 3. 3. Functional Characteristics

This section lists the interface power management feature requirements, defines all the power states a USB interface shall or may implement, and specifies how backward compatibility with USB interfaces that do not implement this feature is maintained.

#### 3.1 Feature Requirements

A USB interface power management feature specification must:

- Define the optional and required power states of a USB interface.
- Maintain backward compatibility with USB interface designs that comply with the *Universal Serial Bus Specification* but do not implement this interface power management feature.
- Have a minimal impact on the *Universal Serial Bus Specification*.
- Define the descriptor information that an interface must provide if it implements this feature.
- Define the requests an interface must respond to if it implements this feature.

#### 3.2 Interface Power States

An interface may implement up to four different, distinct power states. A convention for labeling each optional power state is borrowed from the *Advanced Configuration and Power Interface (ACPI) Specification*.

- x Each power state is given a unique number, 0 through 3. Intended power savings increase with the increase in the state number.
- x Power state numbers are preceded by the letter “D” to indicate transitions between states take place within the device, instead of within the host.

In this specification, the notation ‘Dx’ is occasionally used to refer to the set of states D0, D1, D2, and D3.

The following table summarizes the meaning of the interface power states D0, D1, D2, and D3.

**Table 3–1. D0, D1, D2, and D3 Power State Definitions**

Power State	Power Savings at the Interface	Power Consumption at the Interface	Time to Transition Power State to D0
D0	None	Highest	0
D1	More than D0, but less than D2	Less than D0, but more than D2	Not zero, but faster than D2
D2	More than D1, but less than D3	Less than D1, but more than D3	Slower than D1, but faster than D3
D3	Most	Lowest, may be 0	Slowest



### 3.3 Backward Compatibility with Existing Devices

#### 3.3.1 Host Handling of Devices without Power Descriptors

If the host does not find at least one interface power descriptor in the configuration descriptor that defines the device's current configuration, then the host shall not send any of the Set Feature, Clear Feature, or Get Status request extensions, defined in Section 6, to the device. For more information about the relationship between host interface power management requests and a USB device, see section 4.1.

#### 3.3.2 Device Handling of Request Extensions

The *USB Core Specification* defines a `DEVICE_REMOTE_WAKEUP` feature selector for a Device recipient. This specification allows `DEVICE_REMOTE_WAKEUP` to be used with an Interface recipient.

- When a USB device that implements this interface power management feature receives a Set/Clear Feature (`DEVICE_REMOTE_WAKEUP`) with a Device recipient, then the device shall take the appropriate action (enable or disable remote wakeup) for each interface in the device.

Table 3-2 describes the resulting state of a device and its interfaces after receiving the SetFeature and ClearFeature requests for the `DEVICE_REMOTE_WAKEUP` feature with the device as the recipient. The `DEVICE_REMOTE_WAKEUP` feature state of an interface can be read from the device using a GetStatus request, as described in section 6.2.

**Table 3–2. Power States Resulting from Set/Clear Feature `DEVICE_REMOTE_WAKEUP` Requests for Device Recipient**

Received Request	Device State	Interface State
SetFeature( <code>DEVICE_REMOTE_WAKEUP</code> ) Device is recipient.	Remote Wakeup enabled	Remote wakeup enabled on all interfaces
ClearFeature( <code>DEVICE_REMOTE_WAKEUP</code> ) Device is recipient.	Remote Wakeup disabled	Remote wakeup disabled on all interfaces

Table 3-3 describes the resulting state of a device and its interfaces after receiving the SetFeature and ClearFeature requests for the `DEVICE_REMOTE_WAKEUP` feature with the interface as the recipient. The `DEVICE_REMOTE_WAKEUP` feature state of an interface can be read from the device using a GetStatus request, as described in section 6.2.

**Table 3–3. Power States Resulting from Set/Clear Feature (DEVICE\_REMOTE\_WAKEUP) Requests for Interface Recipient**

Received Request	Device State	Interface State
SetFeature(DEVICE_REMOTE_WAKEUP) Interface is recipient.	Remote Wakeup enabled (because at least one interface has Wakeup enabled)	Remote wakeup enabled for specified interface.
ClearFeature(DEVICE_REMOTE_WAKEUP) Interface is recipient.	Remote Wakeup disabled if all interfaces have remote wakeup disabled.	Remote Wakeup is enabled otherwise Remote wakeup disabled for specified interface.

## 4. Operational Model

This section describes how a device that implements the optional interface power management feature interacts with the host and to events on the USB bus.

### 4.1 How Device Interfaces Respond to Host Power Management Requests

If a device contains one or more interface power descriptors in its firmware, then device logic must be able to respond to the host requests listed in the following table.

**Table 4–1. Device Responses to Host Requests**

Host Request	Device Response	Comment
Standard GetDescriptor (Configuration) request	Return the current Configuration descriptor set (which includes other descriptors besides the Configuration descriptor)	This response is already required of all devices in the USB Core specification. However, the descriptors returned may contain one or more Interface Power Descriptors.
Extended Get Status request, interface is the recipient.	Return the current power state (D0, D1, D2, or D3) of the interface and the current remote wake-up state of the interface (Enabled or Disabled).	For a definition of the interface power management extensions to the standard Get Status request, see section 6.2.
Extended Set/Clear Feature requests, interface is the recipient.	Set the power state of the function to D0, D1, D2, or D3 (as directed by the Set Feature request) and/or set the remote wake-up state of the interface to Enabled or Disabled (as directed by the Set or Clear Feature request).	For a definition of the interface power management extensions to the standard Get Status request, see section 6.2.

### 4.2 General Device Behavior in the Different Power States

An interface can only receive D0, D1, D2, or D3 requests from the host when:

- The host is in the working state (in other words, the host is not in a power saving mode; such host power-saving modes may be referred to by terms such as Host Standby, Host Suspend, or Host Hibernate).
- The USB hub port to which the device or interface is attached is not suspended. After an interface receives a D0, D1, D2, or D3 request and transitions to that power state, it must maintain that state until one of the following events occurs:
- When a request to enter a different power state is received (for example, when the interface is at D3 and it receives a D0 request), the interface must transition to the requested Dx state.
- When a device sees a USB RESET, or receives a SetConfiguration request with the *wValue* field set to 0, the device must transition to the Address state (for a description of the Address device state, see

section 9.1.1 of the *Universal Serial Bus Specification*). Note that in the Address state, the device is not configured.

- When the device receives a SetConfiguration request with the *wValue* field set to a valid non-zero value, all device interfaces transition to D0.
- When the port to which the device is attached loses power, then all device interfaces transition to D3.

### 4.3 How Device Interfaces Respond to USB Bus Power Events

**In general, when the USB enters USB Suspend, all devices on the bus remain in the same Dx state, but must adjust the amount of power they draw from USB.**

The amount of power that a device is allowed to draw depends on the Dx state of all device interfaces, and on whether or not any interfaces are enabled for remote wakeup. Tables 4-2 through 4-5 show the interface power (Dx) states and device power levels that result from Suspend, remote wakeup, Resume and Reset events on the USB bus, for different combinations of bus states, interface Dx states, and interface remote wakeup enable states.

In the tables, the left-most three columns show the bus state, interface Dx state, and interface remote wakeup state when the bus event occurs that is listed in the fourth column of the table. The resulting interface Dx states and device power levels are shown in the right-most two columns of the tables. Note that the Device Power Level in the right-most column is the amount of current the entire device can use (power drawn by the core logic plus the sum of the current drawn by all interfaces on the device).

**Table 4–2. Suspend Event**

Interface Dx State	Bus State	Interface Remote Wakeup State	Bus Event	Resulting Interface Dx State	Resulting Device Power Level
Don't care	Operational	Disabled for all interfaces	Suspend	Does not change	500uA maximum
Any interface at D0	Operational	Don't care	Suspend	Does not change	High-power device configured for high power and enabled as a remote wakeup source can draw up to 2.5mA.  All other devices limited to 500uA.  For more information, see section 7.2.3 of the <i>USB Specification, version 1.1</i>
All interfaces in either D1, D2, or D3	Operational	Enabled for one or more devices	Suspend	Does not change	One Unit load max (100mA)

Table 4–3. Remote Wakeup Event

Interface Dx State	Bus State	Interface Remote Wakeup State	Bus Event	Resulting Interface Dx State	Resulting Device Power Level
All interfaces in either D1, D2, or D3	Suspended	Enabled for one or more interface states	Interface does remote wakeup	Does not change	One Unit load max (100mA). If the device needs more than one Unit load for full operation, it must wait until it gets D0 interface power requests from the host.

Table 4–4. Resume Event

Interface Dx State	Bus State	Interface Remote Wakeup State	Bus Event	Resulting Interface Dx State	Resulting Device Power Level
Don't care	Suspended	Don't care	Resume	Does not change	Transition back to the device power level before the Suspend (the power level of the device immediately before the Suspend event that preceded this Resume).

Table 4–5. Bus Reset Event

Interface Dx State	Bus State	Interface Remote Wakeup State	Bus Event	Resulting Interface Dx State	Resulting Device Power Level
Don't care	Don't care	Don't care	Reset	D0 (all Interfaces and device core logic)	One Unit load max (100mA). The device is not configured.

## 5. Descriptors

This section defines the interface power descriptor. A USB device may implement (a) no interface power descriptor or (b) an interface power descriptor for any interface in a configuration (only one interface power descriptor for each interface).

### 5.1 Interface Power Descriptor

The purpose of an interface power descriptor is to report

- The power states implemented by the interface.
- The ability of the interface to notify the host of external events.

#### 5.1.1 Interface Power Descriptor Placement

An interface power descriptor must be returned after the interface descriptor for the interface to which the interface power descriptor applies, before any endpoint descriptors for that interface, and before the next interface descriptor (if any). For an interface with alternate settings, the interface power descriptor can be returned after any (but only one) of the set of interface descriptors for that interface. If the host software does not find an interface power descriptor, it treats the device as a legacy USB device as defined in the *Universal Serial Bus Specification*.

#### 5.1.2 Interface Power Descriptor Format

The specification of the interface power descriptor follows.

Offset	Field	Size	Value	Description
0	<i>bLength</i>	1	Number	Size of this descriptor in bytes. The value of this field must be 3
1	<i>bDescriptorType</i>	1	Constant	INTERFACE_POWER descriptor type (0x08)
2	<i>bmCapabilitiesFlags</i>	1	Bitmap	Power state and remote wake-up capabilities of the interface.  Bit 0: D0 requests 0 Does not support D0 requests 1 Receives D0 requests  Bit 1: D1 requests 0 Does not support D1 requests 1 Receives D1 requests  Bit 2: D2 requests 0 Does not support D2 requests 1 Receives D2 requests  Bit 3: D3 requests 0 Does not support D3 requests 1 Receives D3 requests  Bit 4: Wake-up capability in D1 0 Cannot signal wake-up from D1 1 Can signal wake-up from D1  Bit 5: Wake-up capability in D2 0 Cannot signal wake-up from D2 1 Can signal wake-up from D2

## 6. Requests

The *Universal Serial Bus Specification* defines a number of standard requests that all devices must support. This section defines how this feature uses those standard requests, if they differ from the standard implementations. This section defines any additional requests defined for this feature.

### 6.1 Set Feature and Clear Feature Selectors

Expanding the feature selectors enables the host to set each interface's power state and to set each interface's remote wake-enable state (enabled or disabled). The additional feature selectors required by the interface power management feature are described in the following table.

Feature Selector	Recipient	Description
DEVICE_REMOTE_WAKEUP.	Interface	Enables the interface as a remote wake-up source when used with Set Feature; disables the interface as a remote wake-up source when used with Clear Feature
INTERFACE_POWER_D0	Interface	Interface transitions to D0; used only with Set Feature.
INTERFACE_POWER_D1	Interface	Interface transitions to D1; used only with Set Feature.
INTERFACE_POWER_D2	Interface	Interface transitions to D2; used only with Set Feature.
INTERFACE_POWER_D3	Interface	Interface transitions to D3; used only with Set Feature.

The values for the new feature selectors are given in the following table.

Feature Selector	Recipient	Value
DEVICE_REMOTE_WAKEUP	Interface	1
INTERFACE_POWER_D0	Interface	2
INTERFACE_POWER_D1	Interface	3
INTERFACE_POWER_D2	Interface	4
INTERFACE_POWER_D3	Interface	5

### 6.2 Get Status Request Extensions

The information returned by a Get Status request to an interface is expanded. The current power state of the interface and the current wake-enable status of the interface are included in the returned information.

This enables the host software to determine the current power state and the current wake-enable state of any interface at any time.

D7	D6	D5	D4	D3	D2	D1	D0
Reserved (reset to zero)				Current power state (0, 1, 2, or 3)		Remote wake-up state	Reserved (reset to zero)
D7	D6	D5	D4	D3	D2	D1	D0
Reserved (reset to zero)							

Bit D1 indicates the remote wake-up state:

- 0 Disabled
- 1 Enabled

Bits D3 and D2 indicate the current power state:

- 00 Power state D0
- 01 Power state D1
- 10 Power state D2
- 11 Power state D3